

CLIENT SIDE LOCALIZATIONS ON THE WORLD WIDE WEB

Technical Field of the Invention

The present invention relates to the field of network browsing software and, in particular, to a method of translating (*i.e.* localizing) a Web page from one language to another language (*e.g.* English to French).

Background of the Invention

In recent years, there has been a tremendous proliferation of computers connected to a global network known as the Internet. A "client" computer connected to the Internet can download digital information from "server" computers connected to the Internet. Client application and operating system software executing on client computers typically accepts commands from a user and obtains data and services by sending requests to server applications running on server computers connected to the Internet. A number of protocols are used to exchange commands and data between computers connected to the Internet. The protocols include the File Transfer Protocol (FTP), the Hyper Text Transfer Protocol (HTTP), the Simple Mail Transfer Protocol (SMTP), and other protocols.

The HTTP protocol is used to access data on the World Wide Web, often referred to as "the Web." The World Wide Web is an information service on the Internet providing documents and links between documents. The World Wide Web is made up of numerous Web sites around the world that maintain and distribute Web documents (*i.e.* otherwise known as Web pages).

A Web site may use one or more Web server computers that are able to store and distribute documents in one of a number of formats including the Hyper Text Markup Language (HTML). An HTML document can contain text, graphics, audio clips, and video clips, as well as metadata or commands providing formatting information. HTML documents also include embedded "links" that reference other data or documents located on the local computer or network server computers.

A Web browser is a client application, software component, or operating system utility that communicates with server computers via FTP, HTTP, and Gopher protocols. Web browsers receive Web documents from the network and present them to a user. Internet Explorer, available from Microsoft Corporation, of Redmond, Washington, is an example of a popular Web browser. As long a Web document is in a computer readable format, such as HTML, a Web browser is able to display the Web document to a user, regardless of the language of the text contained in the Web document (*e.g.* English, French, Italian, German, Spanish, Japanese, etc.). Although the Web browser is able to display the text of the Web document, the text is frequently in a language that is foreign to, and thus unreadable by, the user. This is a significant limitation of the Web.

In the past, Web document and Web site developers have attempted to solve this problem by manually translating each Web document into separate languages. More particularly, developers would create a separate Web page with its own uniform resource locator (URL) for each desired language. Under this approach, each translated Web page is transmitted separately. Thus, this approach requires large volumes of data and increases network transmission. In addition, this approach is very time consuming, costly and inefficient.

Accordingly, it is an object of the present invention to provide a method of allowing users to immediately translate electronic files, such as Web documents, into the language of their choice.

5

Summary of the Invention

The method of the present invention allows users to immediately translate electronic files, such as Web documents, into the language of their choice. In this method, each phrase and/or word in the Web document is assigned an identifier that corresponds to a translation for the phrase. (The present invention works identically for both phrases as well as individual words contained within a Web document. However, only the term "phrase" is referred to hereinafter.) The client is provided a copy of the Web document. A user is then able to select a language in which the user would like the Web document to be displayed. After the user makes his or her language selection, the identifiers in the Web document are used to replace each phrase in the Web document with the foreign language translation of the phrase. The foreign language translation is then substituted for each phrase in the Web document. If a translation for a particular phrase is not available, the particular phrase is not replaced. The resulting Web document is then displayed to the user on a Web browser on the client.

20

Brief Description of the Drawings

The present invention may take physical form in certain parts and steps, embodiments of which will be described in detail in this specification and illustrated in the accompanying drawings which form a part hereof, wherein:

FIGURE 1 is a block diagram of a general-purpose computer system for implementing the present invention;

FIGURE 2 illustrates a Web browser displaying an English language version of a sample Web document prepared in accordance with the present invention;

5 FIGURE 3 depicts the Web browser and Web document of Figure 2, in which the select box of available languages is shown;

FIGURE 4 shows the Web browser displaying a French translation of the Web document depicted in Figure 2;

10 FIGURE 5 illustrates the Web browser displaying a Spanish translation of the Web document depicted in Figure 2; and

FIGURE 6 shows the Web browser displaying a German translation of the Web document illustrated in Figure 2.

Detailed Description of the Invention

15 As illustrated in Figures 2-6, the present invention provides a method of enabling multiple, concurrent, language translation (*i.e.* localization) of Web pages within a Web browser 200. The method allows users to immediately translate prepared pages into the languages of their choice (including double byte character sets) without requiring additional transmissions across the network. In addition, the present invention obviates
20 the current practice of site owners producing separate Web pages for each language to be supported.

In this invention, a client downloads from a server a translation for each phrase contained in a Web page 202. The downloaded phrases are transferred from the server to

the client in the form of a data structure. The data structure can be in an included file or provided by some other mechanism. The present invention supports dynamic changing of languages and the concurrent display of multiple languages. In addition, neither the Web page nor the localized (*i.e.* translated) phrases need to be compiled. Further, each phrase library self-registers itself, and phrases may contain display information (*e.g.* HTML tags) including localized graphics and media files.

Once the phrase translations have been downloaded from a server to the client, a Web browser 200 replaces each phrase in the Web document 202 with its corresponding phrase translation. The resulting Web document 202 is then displayed by the Web browser 200.

In accordance with the present invention, a Web browser 200 executes on a computer, such as a general-purpose personal computer. FIGURE 1 and the following discussion are intended to provide a brief, general description of a suitable computing environment in which the invention may be implemented. Although not required, the invention will be described in the general context of computer-executable instructions, such as program modules, being executed by a personal computer. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Moreover, those skilled in the art will appreciate that the invention may be practiced with other computer system configurations, including hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, and the like. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing

devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

With reference to FIGURE 1, an exemplary system for implementing the
5 invention includes a general purpose computing device in the form of a conventional personal computer 20, including a processing unit 21, a system memory 22, and a system bus 23 that couples various system components including the system memory to the processing unit 21. The system bus 23 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any
10 of a variety of bus architectures. The system memory includes read only memory (ROM) 24 and random access memory (RAM) 25. A basic input/output system 26 (BIOS), containing the basic routines that helps to transfer information between elements within the personal computer 20, such as during start-up, is stored in ROM 24. The personal computer 20 further includes a hard disk drive 27 for reading from and writing to a hard
15 disk, not shown, a magnetic disk drive 28 for reading from or writing to a removable magnetic disk 29, and an optical disk drive 30 for reading from or writing to a removable optical disk 31 such as a CD ROM or other optical media. The hard disk drive 27, magnetic disk drive 28, and optical disk drive 30 are connected to the system bus 23 by a hard disk drive interface 32, a magnetic disk drive interface 33, and an optical drive
20 interface 34, respectively. The drives and their associated computer-readable media provide nonvolatile storage of computer readable instructions, data structures, program modules and other data for the personal computer 20. Although the exemplary environment described herein employs a hard disk, a removable magnetic disk 29 and a

removable optical disk 31, it should be appreciated by those skilled in the art that other types of computer-readable media which can store data that is accessible by a computer, such as magnetic cassettes, flash memory cards, digital versatile disks, Bernoulli cartridges, random access memories (RAMs), read only memories (ROM), and the like, may also be used in the exemplary operating environment.

A number of program modules may be stored on the hard disk, magnetic disk 29, optical disk 31, ROM 24 or RAM 25, including an operating system 35, one or more application programs 36, other program modules 37, and program data 38. A user may enter commands and information into the personal computer 20 through input devices such as a keyboard 40 and pointing device 42. Other input devices (not shown) may include a microphone, joystick, game pad, satellite dish, scanner, or the like. These and other input devices are often connected to the processing unit 21 through a serial port interface 46 that is coupled to the system bus, but may be connected by other interfaces, such as a parallel port, game port or a universal serial bus (USB). A monitor 47 or other type of display device is also connected to the system bus 23 via an interface, such as a video adapter 48. One or more speakers 57 are also connected to the system bus 23 via an interface, such as an audio adapter 56. In addition to the monitor and speakers, personal computers typically include other peripheral output devices (not shown), such as printers.

The personal computer 20 may operate in a networked environment using logical connections to one or more remote computers, such as remote computers 49 and 60. Each remote computer 49 or 60 may be another personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many

or all of the elements described above relative to the personal computer 20, although only a memory storage device 50 or 61 has been illustrated in FIGURE 1. The logical connections depicted in FIGURE 1 include a local area network (LAN) 51 and a wide area network (WAN) 52. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the Internet. As depicted in FIGURE 1, the remote computer 60 communicates with the personal computer 20 via the local area network 51. The remote computer 49 communicates with the personal computer 20 via the wide area network 52.

When used in a LAN networking environment, the personal computer 20 is connected to the local network 51 through a network interface or adapter 53. When used in a WAN networking environment, the personal computer 20 typically includes a modem 54 or other means for establishing communications over the wide area network 52, such as the Internet. The modem 54, which may be internal or external, is connected to the system bus 23 via the serial port interface 46. In a networked environment, program modules depicted relative to the personal computer 20, or portions thereof, may be stored in the remote memory storage device. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

In the present invention, a list of phrases (and/or words) in the HTML page(s) to be localized (*i.e.* translated) is obtained. Each phrase is assigned a unique identifier. Any type of identifier could be used including, but not limited to, ASCII characters, ASCII numbers, alpha-numeric combinations, binary control codes, other numeric

representations, etc. In one embodiment, ASCII numbers are used as the unique identifiers.

After unique identifiers have been assigned, each phrase is localized and then stored in a data structure supported by the Web browser 200. An example of this is shown below. Although this example is written in JavaScript, any scripting or programming language could be utilized. JavaScript is an open scripting language that is supported by many Web browsers, and anyone can use this scripting language without purchasing a license. JavaScript can interact with HTML source code and enables Web authors to design interactive sites that include dynamic content.

[illegible]

In the above example, each language self-registers itself into an array called `languages`, allowing dynamic addition of localization after the page to be displayed is downloaded.

In a typical application, if the user has selected more than one language, then a select box 204, `lcid`, appears on the page displaying the name of the available languages

when the tranonload() procedure is executed. The tranonload() enumerates the available languages 300 in this select box 204.

```
5      <BODY onload=tranonload(>
      <SELECT onchange=TR() class=lcid name=lcid></Select>
      <SCRIPT LANGUAGE=JAVASCRIPT>
      function tranonload(){
          for(i=0;i<languages.length;i++){
              lcid[lcid.length]=new Option(languages[i][0]) //English Title
              lcid[lcid.length-1].value=languages[i][1] //LCID
10          }
      }
```

When the user selects another language, the Web document 202 is preferably searched and phrases are replaced using DHTML. Dynamic HTML (DHTML) refers to Web content that capable of changing each time the Web page is viewed. For example, the same URL could result in a different page depending on any number of parameters, such as: geographic location of the reader, time of day, previous page(s) viewed by the reader, and profile of the reader. Currently, there are many technologies for producing DHTML, including CGI scripts, Server-Side Includes (SSI), cookies, Java, JavaScript, and ActiveX®. Also, DHTML refers to new HTML extensions that enable Web pages to react to user input without sending requests to Web servers. In an alternate embodiment, Web documents are reconstructed using the localized phrases, if DHTML is not available.

As shown below, the identifier for each phrase in the document is located. (In this example, the identifier is part of the class element's name.) After being located, the existing phrase is replaced with its translation—if available. If a translation for a particular phrase is not available, then the original text for that phrase will be maintained and displayed in the Web browser 200.

```
function TR(){  
  i=lcid.selectedIndex // Option (Language) the user selected  
  var vlcid=lcid.options[i].value //get the LCID  
  var ilcid=0  
5   for(i=0;i<languages.length;i++)  
    if(languages[i][1]==vlcicid) //locate the row for the language  
      ilcid=i  
    search=document.all  
    for(i=0;i<search.length;i++){  
10      cn=search[i].className //look at each non-empty class name  
      if (cn.length > 0)  
        { //Look up translation  
          txt=languages[ilcid][2][search[i].className]  
          if (typeof(txt)=="string") //if one exists, replace the text  
15             if (txt.length > 0) search[i].innerHTML=txt  
        }  
      }  
    }  
  }  
}
```

20 In place of a single language translation, translations from several languages may be concatenated and the result displayed instead. Furthermore, the translation may include HTML tags, allowing presentations (formatting, graphics, media files) appropriate for the language to be included. The Source HTML file appears normal except for the addition of a identifier number to the cascading style sheet class name, as
25 shown in the example below.

```
<H1 Class=1> Change Password</H1>  
<P Class=4.label> Enter New Password</P>....  
<DIV Class=4.password>Confirm Password </DIV>
```

30 As previously mentioned, a person of ordinary skill in the art will recognize that there are many other ways to encapsulate the localization identifier in the HTML page.

 The placing of all localized phrases (*i.e.* translations) into a single include-file results in a single file download that may be used by all pages on a Web site with new downloads being required only when the translation-text changes. This may result in
35 smaller file downloads for dynamic pages where the data changes but the text remains the

same (since all of the text may come for the localization file and thus exists in the browser cache). In addition, localization in accordance with the present invention does not require any knowledge of HTML.

A sample implementation of the above-identified invention is shown in Figures 2-6. More particularly, Figure 2 illustrates a Web browser 200 displaying an English language version of a sample Web document 202 prepared in accordance with the present invention. The HTML source code for this Web document 202 (*i.e.* Web page) is as follows.

```
10  <html>
    <head>
    <script LANGUAGE="JSCRIPT">
    var ml=new Array
    var l1=new Array
    var l2=new Array
    15  var l3=new Array
    var l4=new Array
    l1[3]="Identifying the phrases and connecting them to the LP (Localized
    Phrases)requires some human and infrastructure issues to be examined"
    l2[3]="Identifying les expressions et les relier au LP (les expressions localisées)
    20  exige quelques l'humain et issues d'infrastructure à examiner "
    l3[3]="Identifying las frases y conectarlos con el LP (Phrases)requires localizado
    algún ser humano y ediciones de la infraestructura que se examinarán "
    l4[3]="Identifying die Phrasen und Anschließen sie an die Langspielplatte
    (beschränktes Phrases)requires einiger Mensch und zu überprüfende
    25  Infrastrukturausgaben "
    l1[1]='Client Side Example'
    l2[1]='Exemple Latéral De Client '
    l3[1]='Ejemplo Lateral Del Cliente '
    l4[1]='Klient Seitliches Beispiel '
    30  l1[4]="This page illustrates one approach."
    l2[4]="Cette page illustre une approche. "
    l3[4]="Esta paginación ilustra un acercamiento. "
    l4[4]="Diese Seite veranschaulicht eine Annäherung. "
    35  l1[5]="The page is written with 'class=#.{class name}'. This allows the entire
    document to be easily walked and all elements translated (if translation is
    available)"
```

12[5]="La page est écrite avec le ' nom de class=#.{class } '. Ceci permet au document entier d'être facilement marché et tous les éléments d'être traduit (si la traduction est disponible) "

5 13[5]="La paginación se escribe con ' nombre de class=#.{class } '. Esto permite que el documento entero ser recorrida y todos los elementos sea traducido fácilmente (si la traducción está disponible) "

14[5]="Die Seite wird mit ' class=#.{class-Namen } ' geschrieben. Dieses erlaubt, wird daß das gesamte Dokument, leicht gegangen zu werden, übersetzt und alle Elemente (wenn Übersetzung vorhanden ist), "

10 11[11]="Easy automatic extraction of the text into a database (page may be walked from another page, and strings extracted and entered)."

12[11]="Extraction automatique facile du texte dans une base de données (la page peut être marchée d'une autre page, et ficelle extrait et entré). "

15 13[11]="Extracción automática fácil del texto en una base de datos (la paginación se puede recorrer de otra paginación, y encadena extraído y entrado). "

14[11]="Einfache automatische Extraktion des Textes in eine Datenbank (Seite kann von einer anderen Seite gegangen werden, und reiht extrahiert und hereingekommen auf). "

20 ml[1]=l1

ml[2]=l2

ml[3]=l3

ml[4]=l4

lp=l1

25 </script>

<title class="1">Client Side Example</title>

<body>

<h1 class="1">Client Side Example</h1>

30 <select name="lcid" onchange="translate()"

<option value="1">English</option>

<option value="2">French</option>

<option value="3">Spanish</option>

<option value="4">German</option>

35 </select>

<p class="3">Identifying the phrases and connecting them to the LP (Localized Phrases)

requires some human and infrastructure issues to be examined</p>

40 <p><b class="4">This page illustrates one approach.<em class="5"> The page is written

with 'class=#.{class name}'. This allows the entire document to be easily walked and all elements

translated (if translation is available)</p>

<p> Some features are:</p>

45 <li class="10"> Original text is shown if no translation

```
<li class="11"> Easy automatic extraction of the text into a database (page may  
be  
walked from another page, and strings extracted and entered).</li>  
<li class="12"> No events collections are created (as with SPAN or DIV)</li>  
5 <li class="13"> Pages may be easily check by code (and translated)</li>  
</ol>
```

```
<p align="center">I forgot to identify this string on purpose</p>  
<input TYPE="button" onclick="missing()" value="Strings needing translation"  
10 id="button1" name="button1">  
</body>
```

```
<script LANGUAGE="JSCRIPT">  
function missing(){
```

```
  i=lcid.value
```

```
15  if (i>0){
```

```
      lp=ml[i]
```

```
  }
```

```
  search=document.all
```

```
20  for(i=0;i<search.length;i++){
```

```
      //if(search[i].className < 1 && test.length > 0)alert('Not marked for  
translation\n'+test)
```

```
      if(search[i].className > 0){
```

```
          test=search[i].innerHTML
```

```
25          clid=search[i].className
```

```
          txt=lp[clid]
```

```
          if (typeof(txt)!='string')
```

```
              alert('class='+clid+'\nis missing a translation of  
30 \n'+search[i].innerHTML)
```

```
      }
```

```
    }
```

```
  }
```

```
35  function translate(){
```

```
    i=lcid.value
```

```
    if (i>0){
```

```
        lp=ml[i]
```

```
    }
```

```
40  search=document.all
```

```
    for(i=0;i<search.length;i++){
```

```
        if(search[i].className > 0){
```

```
            txt=lp[search[i].className]
```

```
            //alert(typeof(txt))
```

```
45            if (typeof(txt)=="string")
```

```
                if (txt.length > 0) search[i].innerHTML=txt
```

```
}  
}  
</script>  
</html>
```

5

Figure 2 shows the English language text version of this Web document 202 when viewed with a Web browser 200. In this example, English is the default language. A select box 204 is displayed on the Web document 202. This select box 204 identifies the language of the Web document 202 currently displayed. In Figure 2, "English" is identified in the select box 204. Accordingly, the text of the Web document 202 is displayed in the English language.

10

Figure 3 depicts the Web browser 200 and Web document 202 of Figure 2. However, in Figure 3, a user has clicked on (*i.e.* selected) the select box 204. Once selected, the select box 204 displays a list of languages 300 supported by the Web document 202. In this example, the list of languages 300 includes English, French, Spanish and German. Of course, the present invention is dynamic and enables a Web document to support any number of languages.

15

Figure 4 shows the Web browser 200 and Web document 202 of Figure 2, after a user has selected French from the select box 204. After the user selected French, the Web browser 200 searches the Web document 202 for identifiers for each phrase and/or word contained therein. The Web browser 200 then, using the identifiers in the HTML, substitutes the French translation for each phrase and/or word contained in the Web document 202. However, if a translation is not available for any phrase and/or word (in this case French), then the language of the text from the original Web document is maintained. In this example, the Web document was not provided with a French translation for the phrase "Original text is shown if no translation." Accordingly, this

20

25

sentence was displayed in English (*i.e.* the language of the text from the original Web document). Nevertheless, the remainder of the Web document 202 was properly translated into French, as requested by the user.

Figure 5 similarly shows the Web browser 200 and Web document 202 of Figure 2, after a user has selected Spanish from the select box 204. After the user selected Spanish, the Web browser 200 searches the Web document 202 and substitutes the Spanish translation for each phrase and/or word contained therein. Again, if a Spanish translation is not available for any phrase and/or word, then the language of the text from the original Web document is maintained. In this example, the Web document was not provided with a Spanish translation for the phrase "Original text is shown if no translation." Accordingly, this sentence was displayed in English (*i.e.* the language of the text from the original Web document). Nevertheless, the Web browser 200 translated the remainder of the Web document 202 into Spanish, as requested by the user.

Lastly, Figure 6 shows the Web browser 200 and Web document 202 of Figure 2, after a user has selected German from the select box 204. After the user selected German, the Web browser 200 searches the Web document 202 and substitutes the German translation for each phrase and/or word contained therein. As with the other languages, if a German translation is not available for any phrase and/or word, then the language of the text from the original Web document is maintained. The Web document 202 in this example was not provided with a German translation for the phrase "Original text is shown if no translation." Accordingly, this sentence was displayed in English (*i.e.* the language of the text from the original Web document). However, the Web browser

200 translated the remainder of the Web document 202 into German, as requested by the user.

Thus, the present invention has been described in the foregoing specification with reference to specific exemplary embodiments thereof. It will be apparent to those skilled
5 in the art, that a person understanding this invention may conceive of changes or other
embodiments or variations, which utilize the principles of this invention without
departing from the broader spirit and scope of the invention as set forth in the appended
claims. All are considered within the sphere, spirit, and scope of the invention. The
specification and drawings are, therefore, to be regarded in an illustrative rather than
10 restrictive sense. Accordingly, it is not intended that the invention be limited except as
may be necessary in view of the appended claims.